IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 4, 2020, accepted July 9, 2020, date of publication July 20, 2020, date of current version July 30, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3010452

Systematic Evaluation of the Quality Benefits of Spatiotemporal Sample Reprojection in Real-Time Stereoscopic Path Tracing

MARKKU J. MÄKITALO, PETRUS E. J. KIVI, AND PEKKA O. JÄÄSKELÄINEN
Unit of Computing Sciences, Tampere University, 33720 Tampere, Finland
Corresponding author: Markku J. Mäkitalo (markku.makitalo@tuni.fi)

This work was supported in part by the Academy of Finland under Grant 325530, and in part by the ECSEL JU project FitOptiVis under Grant 783162.

ABSTRACT Path tracing is a commonly used but computationally highly expensive stochastic ray tracing method for rendering photorealistic visual content. Combined with a real-time constraint, for example in stereoscopic virtual/augmented reality applications, it typically limits us to rendering at most a few samples per pixel, yielding very noisy results. However, the spatial and temporal redundancies are commonly utilized by reprojecting existing samples between different viewpoints and frames, thus cheaply improving the quality. We provide new insights to the quality benefits of reprojection by systematically evaluating the effective quality of spatiotemporally reprojected stereoscopic path traced data. We show that spatiotemporal reprojection increases the quality of 1 sample per pixel (spp) data by almost a factor of 25 on average, in terms of the effective spp count of the result. Since we are able to reproject 94–98% of the samples, only the remaining 2–6% of the samples in the target frame need to be path traced. We also evaluate how the quality improvement gained through spatiotemporal reprojection scales as the number of input samples per pixel increases, showing that the highest gains are achieved at the lowest input spp counts. Finally, we show how blending existing path traced data and stereoscopically reprojected data further improves the quality of spatiotemporal reprojection, on average yielding a 47% higher effective spp than without blending.

INDEX TERMS Computer graphics, path tracing, ray tracing, real-time rendering, stereoscopic.

I. INTRODUCTION

Ray tracing is a rendering technique that simulates how light travels and interacts with objects in the real world. In the past decade, ray tracing has gained a major increase in mainstream popularity, first driven by offline applications in the motion picture industry, and then followed by the real-time applications in the gaming industry and the domains of virtual and augmented reality (VR/AR). This development has been possible due to recent advances in both hardware and rendering software, making it feasible to render photorealistic scenes with reasonable computational resources, and within a reasonable amount of time.

Nevertheless, modern flavours of ray tracing, most commonly path tracing, typically rely on stochastic procedures to approximate the global illumination, in other words the integral *rendering equation* [1]. This approach implies a tradeoff between rendering time and rendering quality. For instance, path tracing typically requires tracing thousands of samples per pixel (spp) in order to reduce the approximation error, i.e., the noise, to an imperceptible level; even at 1024 spp, there still tends to be a small amount of visible noise [2]. Even with modern consumer hardware that provides hardware acceleration for ray tracing, in high-resolution real-time applications we are limited to tracing only a few samples per pixel. Hence, the rendered raw pixels are very noisy and must go through a series of post-processing steps in order to attain visually acceptable quality. Evaluating this quality improvement in a rigorous manner is also an interesting question on its own, and it is the focus of this paper.

VR/AR applications also commonly impose an additional challenge: for a stereoscopic view, a separate viewpoint is rendered for each eye, thus intuitively doubling the amount of pixels to be rendered. By extension, even more data are

The associate editor coordinating the review of this manuscript and approving it for publication was Gangyi Jiang.

needed for more complex multi-view applications. However, these views are generally partially overlapping, which means there is significant redundancy between the different viewpoints. Moreover, in both synthetically created and naturally captured imagery, such redundancy is commonly found even within a single viewpoint (spatial self-similarity), and especially between successive frames as the camera or the objects move (temporal coherence).

A common computationally inexpensive post-processing step that takes advantage of this redundancy is known as reprojection. It means we use already rendered samples and reproject them into new locations based on the known camera movement and possibly other auxiliary data, such as the depth map and shading normals of the scene; in the case of a synthetic scene, these are usually readily available and noise-free. The reprojection can be done both spatially, i.e., from one viewpoint to another at the same time instant, and temporally, i.e., from previous frame(s) to the next. Reprojection not only reduces the amount of data that need to be ray traced, but it also acts as a simple denoising filter.

Even though reprojection is commonly used, and typically justified through its computational savings, its quality benefits are often overlooked and not rigorously quantified. Especially in real-time path tracing, where the data are very noisy, it is important to understand how reprojection affects the quality of the data. We provide a systematic quality evaluation through the *effective sample per pixel* count of the reprojected data. In other words, we evaluate how many samples per pixel would need to be path traced so that the error is equal to that of the reprojected frame. To the best of our knowledge, no systematic quality evaluations have been conducted with this metric before us.

This work is a continuation of our preliminary work on evaluating the quality benefits of spatiotemporal reprojection [3] for path traced 1 spp input data. In this paper, we provide more comprehensive results, evaluate how the quality of reprojected data scales with higher than 1 spp inputs, and consider also stereoscopic blending of existing path traced data and reprojected data.

The rest of the paper is organized as follows: Section II introduces the basics of temporal reprojection and stereo reprojection, and discusses related work. Section III focuses on how reprojection improves the quality of noisy data, and on how we measure the quality improvement objectively. Section IV systematically evaluates the quality improvement in various scenarios in terms of the effective spp of the result, for stereo and temporal reprojection both separately and combined. Finally, Section V presents our conclusions.

II. BACKGROUND AND RELATED WORK

A. TEMPORAL REPROJECTION

The concept of temporal coherence, that the contents of successive frames do not usually change significantly, can be traced back to at least the early 1970s, when it was discussed in conjunction with visibility determination [4]. The following decade saw several breakthroughs in the field of ray tracing, with seminal works about recursive ray tracing [5], distributed effects such as motion blur and depth of field [6], and Monte Carlo style stochastic ray tracing (*path tracing*) that computes global illumination through the rendering equation [1]. Soon after, an algorithm for using temporal coherence to accelerate ray tracing for motion picture production was introduced in [7]. Specifically, it gathered object space information from the previous frame and then estimated where the objects would be in the current frame, thus performing *forward* sample reprojection. This resulted in having to trace less than 40% of the pixels in the current frame. The forward reprojection approach was generalized in [8], where it was reported to yield up to 92% savings in rendering time.

In [9], a render cache was introduced, which enabled interactive ray tracing for low resolutions (about 8 frames per second for a 320×320 resolution). It builds an acceleration structure by caching previously rendered samples, storing their colours and also the 3D data and shading information. That allowed for various heuristics to be used when reprojecting the samples onto the new frame, such as comparing the depth data and colour contrast for detecting holes, disocclusions, and other artifacts [10]. Whereas the original render cache still used forward reprojection, its concept of storing the earlier data paved way for a *backward reprojection* cache (also known as reverse reprojection), introduced independently in [11] and [12].

Backward reprojection works by starting from the current frame to be rendered instead of an earlier frame, and for each pixel in the current frame determining its location in the earlier frame. If the pixel was visible in the earlier frame and thus was stored in the cache, it can potentially be reprojected onto the current frame. Various heuristics can also be used here in deciding whether to ultimately reproject the found pixel or not, for instance based on the depth values or surface normals. Even though this backward mapping simplifies the reprojection, it does bring additional requirements in terms of storing and handling past data in memory.

Nowadays reprojection is used by modern real-time path tracing reconstruction filters [13]–[16], in order to generate a better quality input before the actual denoising or reconstruction filter. Moreover, reprojection is used extensively in rasterized game graphics. One of the most commonly used reprojection methods is called Temporal Anti-Aliasing (TAA) [17], which generates an anti-aliased image without extra spatial samples. Instead, the temporal samples are reprojected and used for smoothing the edges in the image. With TAA, the camera is typically sub-pixel jittered with a Halton sequence for achieving the same smoothing effect even with a static camera. For more details on TAA, we refer to the recent survey [18]. However, note that TAA can introduce an excessive amount of blur. This can be mitigated through, e.g., a combination of reprojection and superresolution bitmasks, achieving temporal stability while maintaining sharp features [19].

For further details and history about temporal coherence and reprojection, as well as for reviewing a multitude of other ways of accelerating ray tracing to an interactive/real-time level, we refer the reader to [10] and [20]. Reconstruction algorithms for Monte Carlo rendering are also surveyed in [21].

B. STEREO REPROJECTION

In stereo reprojection, we reproject samples from one spatial viewpoint to another at the same time instant. In the standard stereoscopic case, the eyes can be thought of as two distinct cameras that are separated by approximately 6.5 centimeters (the eye separation of an average human), and we reproject samples from one camera's viewpoint to the other.

There are two well-established methods for setting up stereoscopic cameras: a *parallel* or *sensor-shift* camera setup, where the two virtual cameras are translated only horizontally, and a *converged* or *toe-in* camera setup, which additionally introduces a slight inward rotation for convergence. Converged camera setups have been shown to produce visual distortions such as *keystoning* [22], and thus parallel stereoscopic cameras are preferred for viewing comfort [23].

Stereo reprojection can be thought of as a special case of temporal reprojection [24]. Specifically, if there is only camera movement and the scene remains otherwise static, there is no fundamental difference on whether the camera movement is interpreted as a change in viewpoint spatially or temporally.

An early example of stereo reprojection [25] built on the work of [7] and combined it with a calculation of stereo disparity information in order to do the reprojection. With their method, between 20% and 50% of the pixels in the target frame needed to be ray traced after reprojection. A more optimized method [24] made a simplifying observation that they could only reproject the x-coordinates of the samples, based on an assumption that the observer's eyes are level (i.e., their head is not tilted). They obtain an estimate of 93% reduction in the amount of rays that need to be traced for the second eye, albeit they do not address all problems related to the disparity between the two views [26].

A more general approach that does not assume horizontally level eyes, introduced in [26], leveraged coherence in the epipolar geometry of a stereo image pair, subdividing the space with epipolar planes; their algorithm ran in 30–50% of the time of their comparison algorithm.

Reprojection based methods can also support animated rigid objects [27]. If there is a way to compute the screen space motion vector for an animated object, it can be both forward and backward reprojected. The motion vectors can be computed for common rigid body animations such as translation, rotation and scaling.

With the modern rise in popularity of 3D content, stereoscopic ray tracing and reprojection algorithms have also evolved to cover more general multi-view rendering; see, e.g., [28], [29] and the references therein. However, as even monoscopic real-time ray tracing is only now becoming tractable, enabling high quality stereoscopic real-time ray tracing remains a challenge for the near future. The recent NVIDIA Turing architecture offers hardware acceleration for rendering up to four views in a single render pass, allowing discrepancy between the eyes also in the y-coordinate, whereas the acceleration introduced in their earlier Pascal architecture only supported two views (implying a narrower field of view) and discrepancy only in the x-coordinate, assuming horizontally level eyes [30]. However, its potential in combination with their ray tracing hardware is not detailed.

As seen above, the advantage of sample reprojection is typically expressed in terms of saved rendering time or the amount of skipped rays. However, especially in path tracing where the error is progressively reduced by aggregating noisy samples, it is also important to understand how reprojection affects the quality of the final result. Hence, this paper focuses on systematically evaluating the advantages of reprojection in terms of quality.

III. EVALUATING THE QUALITY BENEFITS OF REPROJECTION

Sample reprojection is particularly relevant for real-time path tracing because of the strict time budget available for rendering each frame. Applying reprojection and temporal accumulation is roughly an order of magnitude faster than real-time path tracing [13], [31], so it is a cheap way to increase the quality of the rendered result without heavy computational overhead.

Even though the overlap between adjacent viewpoints is usually significant, not all pixels can necessarily be reprojected from one viewpoint to another. This is typically due to occlusions, non-diffuse shading, and the general non-bijective nature of the reprojection mapping (i.e., many pixels may map onto a single pixel in another viewpoint, and some pixels may not have any pixel mapped to them from another viewpoint) [9]. The backward reprojection approach can be used to mitigate the problems caused by non-bijectivity.

As we observed in [3], it is useful to evaluate the quality in terms of the effective sample per pixel count, i.e., how many path traced samples would be needed in order to attain an error equal to that of the reprojected frame. This approach is rarely used, perhaps due to the fact that real-time path tracing has only recently become practically feasible; in offline scenarios, there is not such a stringent need to compromise the quality in favour of speed, even if reprojection is used to accelerate the rendering.

In practice, these errors can be computed for example through the root mean square error (RMSE), or the structural similarity index (SSIM) [32], between a noisy frame and the corresponding noise-free reference frame path traced with 4096 spp. Other error metrics can also be used, assuming the obtained error value decreases monotonically as the number of input samples per pixel increases; otherwise, the definition of the effective spp would be ambiguous.

If reprojection is done from a path traced viewpoint onto a target viewpoint with no existing samples (e.g., from the right eye to the left), we need to path trace only the missing pixels in the target viewpoint where reprojected data could not be utilized. On the other hand, we may have path traced data in both viewpoints, on top of which we blend reprojected data with a chosen ratio. In the latter case, we can also simultaneously *cross-reproject* samples from each viewpoint to the other without introducing excessive blur; in the former case, first reprojecting one way and then re-reprojecting back to the original viewpoint would apply two separate blurring kernels on the same data. Note that the blurring occurs because the reprojection generally results in fractional pixel locations, so resampling is required. The resampling is typically done via bilinear or bicubic filtering [18]. Nevertheless, a reasonable amount of blur is still advantageous, as it acts as a simple denoising filter and usually improves the quality of the result, as we see in Section IV.

In temporal reprojection, samples are commonly accumulated for temporal stability. The accumulated samples are reprojected onto the new frame, and then blended with new samples according to some blending factor α. The common approach of using a constant α results in an exponential smoothing filter, also known as exponential moving average, assigning progressively lower weights to the older samples that get less relevant [18].

IV. EXPERIMENTS AND RESULTS

A. EXPERIMENTAL SETUP

In our experiments, we consider both stereoscopic scenarios outlined in Section III. Specifically, in Section IV-B, we reproject n spp pixels spatially from a path traced right-eye viewpoint into the empty left-eye viewpoint, hence no stereo blending. In Section IV-C, we spatially cross-reproject n spp path traced pixels between each eye's viewpoint and blend the reprojected data with the existing path traced n spp data with a 50-50 ratio, i.e., the reprojected and path traced data will be combined with equal weights.

In both cases, we also apply TAA-style temporal reprojection (without sub-pixel jittering), accumulating the pixels using the exponential moving average with a constant blending factor $\alpha = 0.2$, In other words, we weigh the existing accumulated data with a factor of 0.8, and combine them with the newest reprojected frame with a weight factor of 0.2. We chose the value $\alpha = 0.2$, as it has been shown to yield good results in, e.g., [15] and [13].

Both spatial and temporal reprojection are done with the backward reprojection approach explained in Section II, using bilinear interpolation between the four samples closest to the fractional reprojected pixel location; this is further illustrated in Fig. 1. We discard the reprojected sample if the difference in the depths, 3D world positions or shading normal values between the source and the target location is deemed too large, or if the reprojection location goes out of the frame and data are thus not available. The limits of acceptable differences are set manually based on the scale of each test scene. The samples missing due to the discarding process are then finally path traced according to the input spp n.

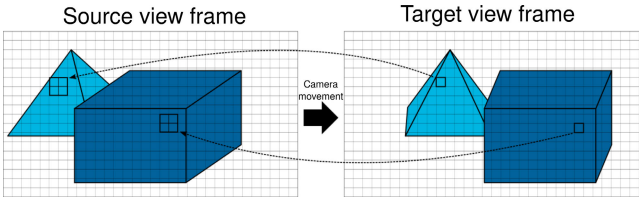


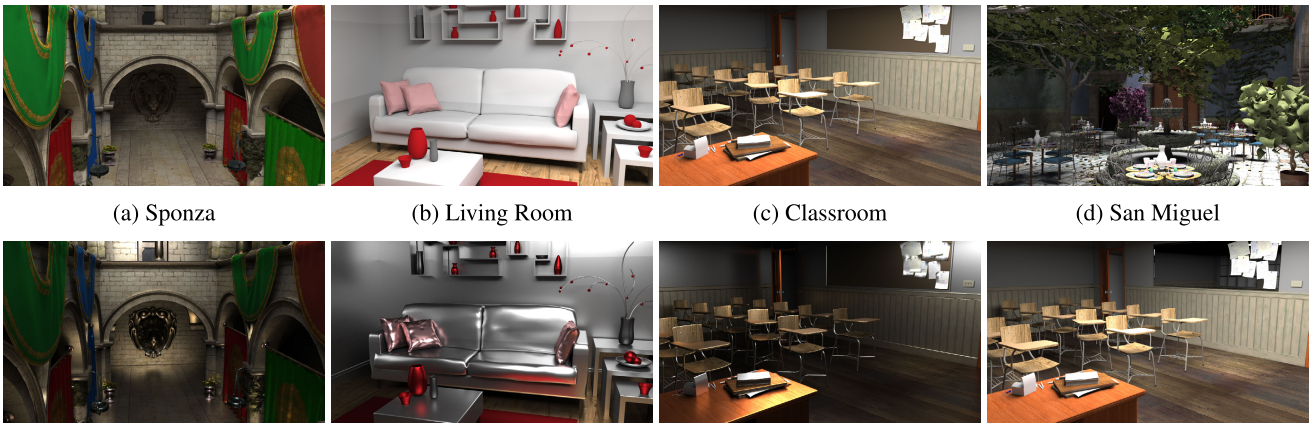
FIGURE 1. Backward reprojection with bilinear interpolation. The camera movement can be in space (stereo reprojection) or in time (temporal reprojection).

The quality of the reprojected data is evaluated in terms of the *effective spp* as explained in Section III, through the SSIM error metric. For the error value obtained for each frame, we find the purely path traced m and $m + 1$ spp comparison frames, between whose corresponding error values the obtained error lies, and interpolate the effective spp based on that. In particular, we compute the SSIM over the RGB image as a 3D volume. SSIM can also be computed over the luminance channel of the image, but with the latter we observed nonmonotonic behaviour in certain cases, leading to ambiguities in computing the effective spp, as explained in Section III. In well-defined cases, the results obtained with both flavours of SSIM are generally comparable.

We evaluate the effective spp in order to quantify the improved quality gained through spatial reprojection, temporal reprojection, and the combination of the two. Moreover, we investigate how this quality improvement scales with respect to the quality of the original input. More specifically, we conduct the experiments for path traced input data having $n = 1, 2, 4, 8, 16, 32, \ldots, 512$ spp in the source viewpoint(s), with a particular focus on $n = 1, 2, 4, 8, 16, 32$ spp. The latter range covers the input spp counts that can be currently path traced in real time for modern high-resolution applications, and also anticipates what is likely to be achievable in the near future with the next generations of real-time rendering hardware.

Our test scenes are created using the Sponza, Living Room, Classroom, and San Miguel scenes. We render versions of them with all-diffuse materials, all-glossy materials, a mixture of diffuse and reflective materials, and moving light sources. Each rendered stereoscopic scene has 60 frames path traced with a 1280×720 resolution for each eye, and a moving observer (i.e., moving camera). Fig. 2 shows example 4096 spp reference frames of the test scenes.

Finally, we note that in path tracing each sample, we always trace one primary ray, one secondary ray, and two shadow rays; this is a realistic amount of data to be traced in real time. Even when reprojecting into an empty viewpoint, we still assume we have access to the G-buffers (depths, world positions, and shading normals) for both viewpoints, in order to determine when reprojected data should be discarded. This assumption is reasonable, as rendering these auxiliary buffers



(a) Sponza (b) Living Room (c) Classroom (d) San Miguel

(e) Sponza (Glossy) (f) Living Room (Glossy) (g) Classroom (Glossy) (h) Classroom (Mirror)

FIGURE 2. 4096 spp references of the scenes used in the experiments. Scenes (a) and (b) are also used as a base for Sponza (Moving Light) and Living Room (Moving Light), respectively. In (h), the diffuse notice board of (c) has been replaced with a mirror.

is computationally inexpensive compared to actually path tracing the samples.

B. REPROJECTION WITHOUT STEREO BLENDING

Here we consider the scenario where the source viewpoint (right eye) is fully path traced, and the target viewpoint (left eye) does not yet contain any data. Then we apply backward sample reprojection for each pixel in the target viewpoint.

When evaluating the combined effect of spatial and temporal reprojection, we first apply temporal reprojection to the right eye, after which the stereo reprojection is done from the right eye to the left eye. Then, the missing samples in the left eye are path traced, and the resulting left-eye quality is evaluated. We assume temporal accumulation is also done for the left-eye data.

1) EFFECTIVE SPP AND ITS SCALABILITY

Tables 1–3 present the effective reprojected spp counts for $n = 1, 2, 4, 8, 16, 32$ spp input data after stereo reprojection, temporal reprojection, and spatiotemporal reprojection, respectively. Each average is taken over the last 50 frames in the scene, and the standard deviation of the effective framewise spp counts is also presented. In addition, the effective spp averaged across all scenes is reported, along with its standard deviation.

Note that the first 10 frames are omitted while computing the averages and standard deviations, as they are not representative of the overall long-term quality. More specifically, with the chosen temporal blending ratio, it takes about 5–10 frames until enough data have been accumulated in order to stabilize the quality; such initial frames can typically be hidden from the user in practical applications. Fig. 3 demonstrates this behaviour through an example of the framewise SSIM error values, for the 1 spp input Classroom (Mirror) scene.

As can be seen from Table 1, stereo reprojecting 1 spp data into an empty viewpoint yields an effective 1.69 spp quality

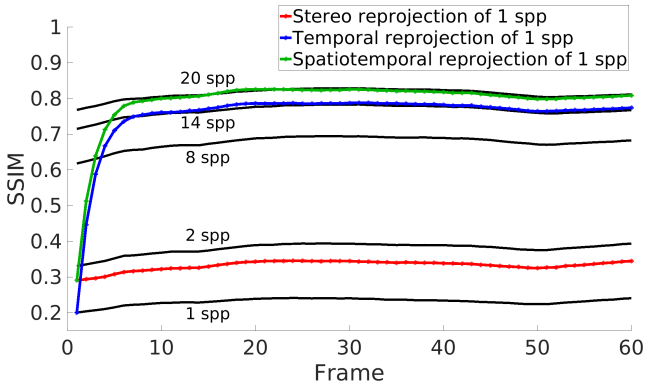


FIGURE 3. Classroom (Mirror) scene: Framewise SSIM values of the reprojected 1 spp frames (without stereo blending), and of comparison data having 1 spp, 2 spp, 8 spp, 14 spp, and 20 spp.

in the target eye, averaged across all scenes. In other words, the denoising happening through bilinear interpolation in the reprojection increases the quality of the 1 spp data by an additional 69%. Moreover, the quality is stable across individual frames and across all scenes; the variations are relatively minor.

As for higher-spp inputs, the average effective spp is 3.19 spp for 2 spp input, 6.19 spp for 4 spp input, 12.12 spp for 8 spp input, 23.71 spp for 16 spp input, and 46.03 spp for 32 spp input. Overall, the quality of the result scales almost linearly over the 1–32 spp input range, as is further illustrated in Fig. 4. However, with the figure showing the full 1–512 spp input range, we also see that the gains for most of the test scenes diminish as the input spp count increases to the order of hundreds, sometimes even resulting in a worse effective spp than the input spp. This can be explained by the bilinear interpolation beginning to be a dominating factor instead of the path tracing noise, causing excessive blur to the already reasonably low-noise input image. These cases are visible in the figure as data points below the "equal quality" line.

TABLE 1. Effective spp after stereo reprojection only (without stereo blending), as measured by SSIM. The average spp for each scene is taken over 50 frames, with the respective standard deviation also shown.

	1 spp		2 spp		4 spp		8 spp		16 spp		32 spp	
	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std
Sponza	**1.69**	0.02	**3.21**	0.07	**6.14**	0.15	**11.95**	0.36	**23.38**	0.85	**45.19**	1.90
Sponza (Glossy)	**1.73**	0.03	**3.32**	0.08	**6.42**	0.22	**12.72**	0.59	**25.22**	1.26	**49.54**	2.55
Sponza (Moving Light)	**1.68**	0.03	**3.20**	0.11	**6.17**	0.26	**12.09**	0.54	**23.71**	1.22	**46.17**	3.07
Living Room	**1.69**	0.01	**3.15**	0.03	**6.18**	0.05	**12.28**	0.09	**24.42**	0.18	**48.51**	0.39
Living Room (Glossy)	**1.80**	0.01	**3.47**	0.04	**6.83**	0.10	**13.36**	0.22	**26.05**	0.39	**50.79**	0.63
Living Room (Moving Light)	**1.68**	0.02	**3.15**	0.04	**6.21**	0.07	**12.33**	0.10	**24.49**	0.19	**48.59**	0.40
Classroom	**1.69**	0.01	**3.10**	0.02	**5.96**	0.03	**11.58**	0.08	**22.38**	0.17	**42.65**	0.49
Classroom (Glossy)	**1.72**	0.01	**3.20**	0.03	**6.17**	0.07	**11.99**	0.12	**23.24**	0.21	**44.75**	0.40
Classroom (Mirror)	**1.68**	0.01	**3.08**	0.03	**5.90**	0.04	**11.40**	0.10	**21.87**	0.28	**41.16**	0.67
San Miguel	**1.53**	0.01	**2.98**	0.03	**5.87**	0.06	**11.49**	0.14	**22.32**	0.32	**42.94**	0.67
Scene Average ± Std	**1.69** ± 0.07		**3.19** ± 0.13		**6.19** ± 0.28		**12.12** ± 0.60		**23.71** ± 1.34		**46.03** ± 3.24	

TABLE 2. Effective spp after temporal reprojection only (without stereo blending), as measured by SSIM. The average spp for each scene is taken over 50 frames, with the respective standard deviation also shown.

	1 spp		2 spp		4 spp		8 spp		16 spp		32 spp	
	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std
Sponza	**15.81**	0.63	**27.43**	1.22	**47.50**	2.53	**80.45**	5.04	**129.52**	9.19	**194.45**	14.92
Sponza (Glossy)	**23.81**	1.29	**45.52**	2.62	**84.90**	5.02	**154.58**	10.08	**268.58**	17.65	**453.03**	30.78
Sponza (Moving Light)	**14.98**	0.66	**25.98**	0.92	**44.95**	1.94	**75.91**	4.60	**122.19**	10.24	**184.62**	21.23
Living Room	**16.41**	0.37	**31.43**	0.76	**60.95**	1.63	**118.09**	3.43	**225.96**	6.97	**423.97**	14.43
Living Room (Glossy)	**29.55**	0.60	**50.65**	1.12	**87.41**	1.71	**153.20**	3.13	**271.96**	5.00	**486.74**	9.04
Living Room (Moving Light)	**15.70**	0.73	**29.78**	0.93	**57.40**	1.28	**110.52**	1.92	**210.35**	3.35	**390.62**	6.82
Classroom	**14.73**	0.25	**26.54**	0.59	**47.46**	1.53	**82.30**	3.79	**135.75**	8.78	**207.72**	17.61
Classroom (Glossy)	**18.63**	0.76	**32.64**	1.43	**56.94**	2.64	**97.67**	4.90	**161.86**	8.92	**253.49**	14.87
Classroom (Mirror)	**14.61**	0.22	**26.23**	0.54	**46.66**	1.59	**80.18**	4.31	**130.68**	10.17	**197.23**	20.51
San Miguel	**22.02**	0.51	**33.73**	1.12	**53.07**	2.56	**85.01**	5.29	**137.13**	10.11	**217.79**	17.70
Scene Average ± Std	**18.62** ± 4.99		**32.99** ± 8.50		**58.72** ± 15.41		**103.79** ± 29.82		**179.40** ± 59.42		**300.97** ± 122.16	

TABLE 3. Effective spp after spatiotemporal reprojection (without stereo blending), as measured by SSIM. The average spp for each scene is taken over 50 frames, with the respective standard deviation also shown.

	1 spp		2 spp		4 spp		8 spp		16 spp		32 spp	
	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std
Sponza	**20.03**	1.32	**34.19**	2.62	**57.32**	5.35	**92.08**	10.16	**138.72**	17.11	**193.27**	25.89
Sponza (Glossy)	**31.86**	3.06	**60.04**	5.90	**109.72**	10.44	**191.01**	19.60	**322.48**	33.36	**518.36**	56.15
Sponza (Moving Light)	**18.84**	0.97	**32.36**	2.09	**54.57**	5.11	**88.51**	11.32	**135.09**	22.33	**192.32**	40.34
Living Room	**23.21**	0.64	**44.43**	1.32	**85.21**	2.83	**162.70**	5.97	**304.91**	12.23	**557.14**	23.40
Living Room (Glossy)	**41.58**	1.07	**70.56**	1.53	**121.18**	2.78	**210.64**	5.00	**370.93**	8.94	**649.30**	17.33
Living Room (Moving Light)	**22.20**	0.94	**42.17**	1.28	**80.36**	1.92	**152.51**	3.40	**283.68**	6.58	**512.80**	12.11
Classroom	**19.99**	0.41	**35.29**	0.87	**60.44**	1.90	**98.11**	4.31	**147.16**	8.47	**200.75**	14.25
Classroom (Glossy)	**24.95**	0.95	**43.00**	1.75	**72.50**	2.87	**118.58**	4.53	**183.12**	7.60	**261.21**	11.40
Classroom (Mirror)	**19.43**	0.51	**33.90**	1.26	**57.04**	3.03	**90.53**	7.09	**132.48**	13.83	**176.28**	22.59
San Miguel	**26.30**	0.66	**40.66**	1.26	**63.93**	2.71	**101.12**	5.65	**157.96**	10.46	**236.47**	17.54
Scene Average ± Std	**24.84** ± 7.09		**43.66** ± 12.41		**76.23** ± 23.21		**130.58** ± 45.33		**217.65** ± 92.19		**349.79** ± 185.59	

Table 2 shows that temporal reprojection and accumulation is very effective in increasing the quality of the result, and also that the amount of improvement depends substantially on the complexity of the scene. Moreover, the minimum effective spp is always equal to the input spp, because the first frame does not have any temporal accumulation; the quality then quickly increases and stabilizes, as discussed above.

The results after combining both reprojection methods into the final spatiotemporal reprojection are presented in Table 3: for 1 spp input, the average effective spp is between 18.8–41.6 spp, 32.4–70.6 spp for 2 spp input, 54.6–121.2 spp for 4 spp input, 88.5–210.6 spp for 8 spp input, 132.5–370.9 spp for 16 spp input, and 176.3–649.3 spp for 32 spp input. Thus, even in the worst case, we get an almost 19-fold increase in quality for 1 spp input data, with significantly less computational effort than it would take to path trace 19 samples per pixel; on average, the improvement is almost 25-fold.

The scalability as a function of input spp is also shown in Fig. 5 for the full 1–512 spp input range. Even though especially the Living Room scenes still greatly benefit from the reprojection even at a high input spp count, possibly due to their lack of complex geometry and significant occlusions, the general tendency is again that the improvements saturate after a certain point. This illustrates that sample reprojection

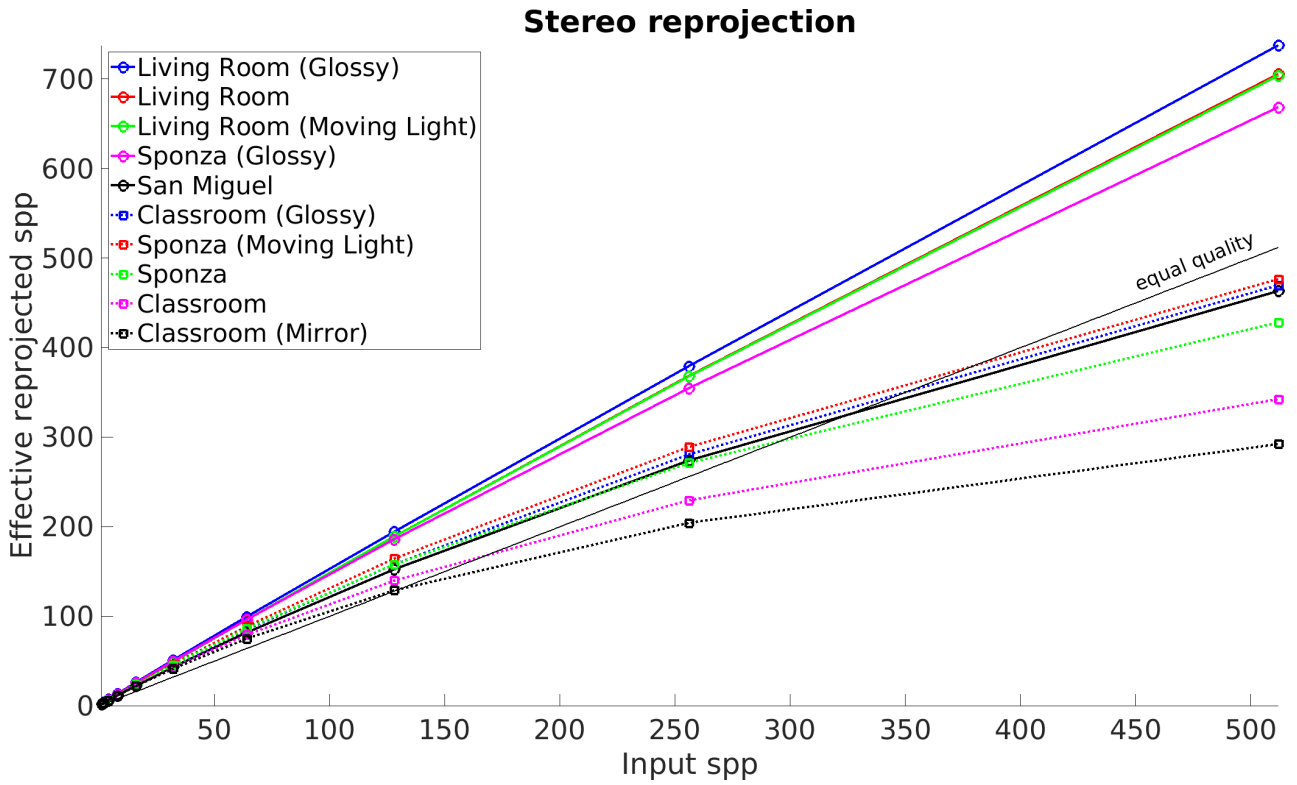


FIGURE 4. Average effective spp counts after stereo reprojection (without stereo blending) as a function of input spp. The results for 1–32 spp are also presented in Table 1.

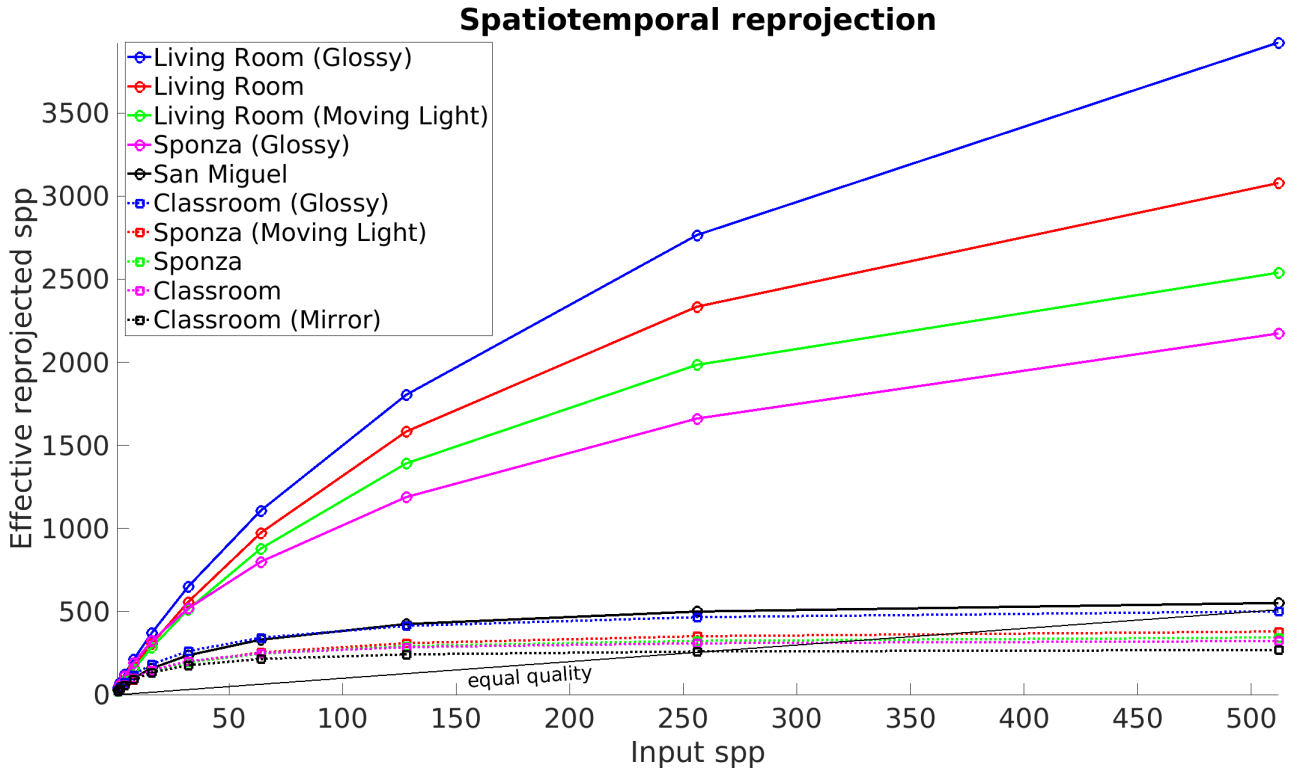
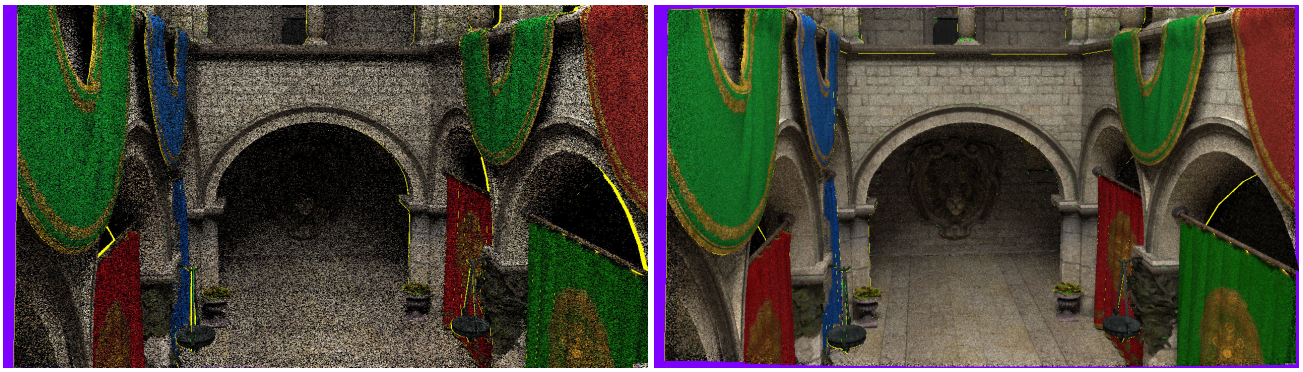


FIGURE 5. Average effective spp counts after spatiotemporal reprojection (without stereo blending) as a function of input spp. The results for 1–32 spp are also presented in Table 3.



(a) Stereo reprojected 1 spp data (b) Temporally reprojected and accumulated 1 spp data

FIGURE 6. Pixels discarded by the reprojection algorithm in (a) stereoscopically reprojected and (b) temporally reprojected and accumulated path traced images: bright yellow for discards based on world position or depth, bright green for discards based on shading normals, and purple for discards based on pixels reprojected outside of the frame. Spatiotemporal reprojection combines the results of (a) and (b).

TABLE 4. Percentage of pixels discarded during stereo and temporal reprojection, with discarding being done based on depth, world position and shading normal differences. The average, minimum and maximum for stereo (60 frames) and temporal (59 frames) discard percentages are shown.

	Stereo		
	Avg	Min	Max
Sponza (All scenes)	**1.94**	1.19	2.89
Living Room (All scenes)	**2.47**	2.30	2.67
Classroom (All scenes)	**4.36**	3.61	5.84
San Miguel	**5.86**	5.07	12.20
	Temporal		
	Avg	Min	Max
Sponza (All scenes)	**3.96**	3.57	4.47
Living Room (All scenes)	**1.49**	0.71	2.28
Classroom (All scenes)	**2.22**	1.56	2.99
San Miguel	**3.58**	2.41	6.09

is typically the most beneficial for data with low spp counts.

2) DISCARD PERCENTAGES

Table 4 presents the percentages of pixels discarded during stereo reprojection and temporal reprojection, further corroborating the amount of saved computational effort: on average, only about 2–6% of the pixels could not be reprojected, meaning only 2–6% of the pixels had to be path traced for the target viewpoint. More specifically, the average discard percentages are about 2–4% for all other scenes, except for San Miguel it is close to 6% due to the detailed foliage and other intricate geometry. Fig. 6 visualizes the discarded pixels for an example frame of Sponza.

3) VISUAL QUALITY

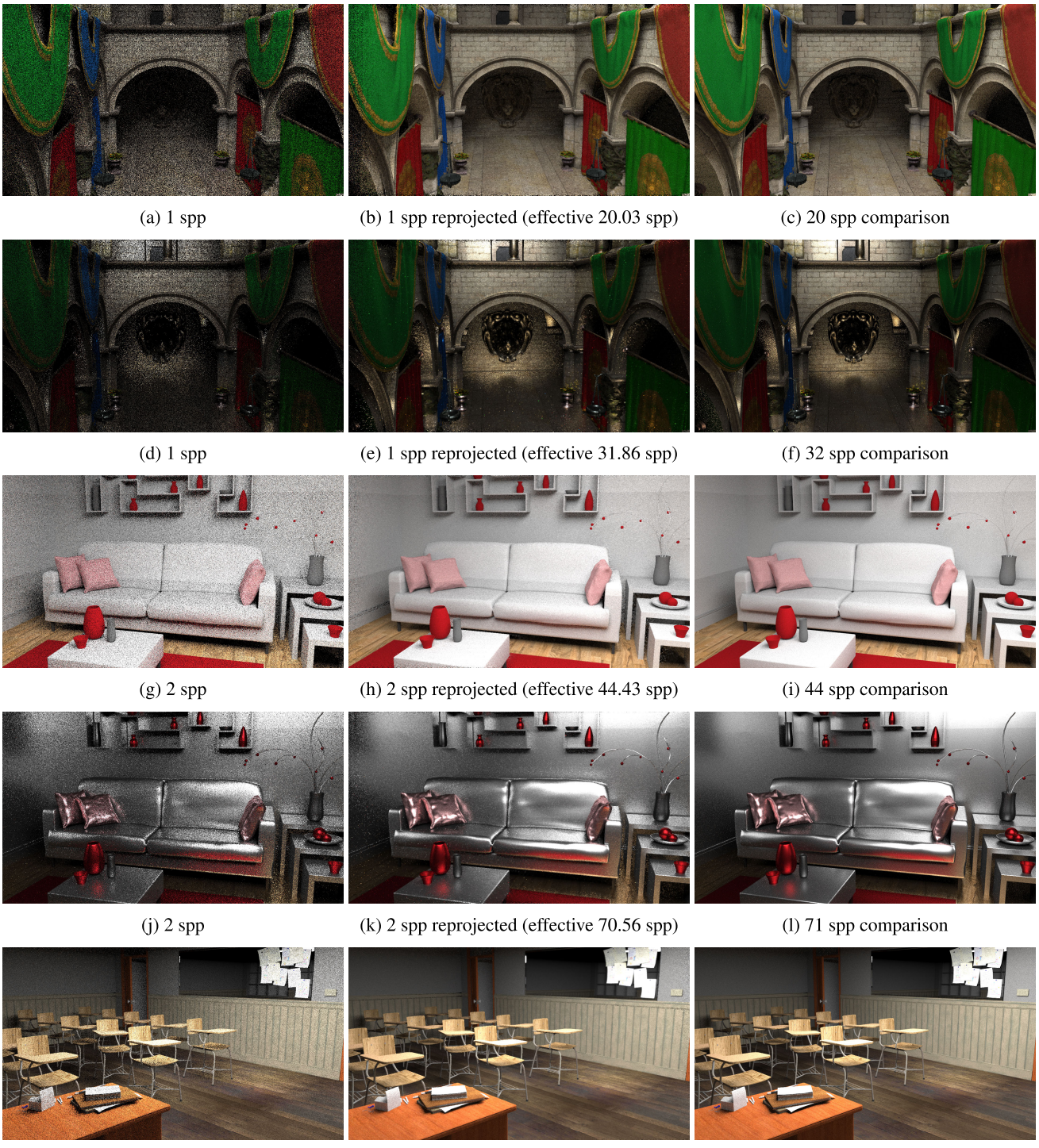
Fig. 7 demonstrates the visual quality of the obtained results for several scenes: The left column shows an example left-eye frame with purely path traced n spp, $n \in \{1, 2, 4, 8\}$. The middle column shows the same left-eye frame that was instead obtained by spatiotemporally reprojecting the corresponding n spp right-eye frame onto an empty left eye and path tracing only the missing samples, with temporal accumulation also applied. Finally, the right column shows a purely path traced comparison, with the spp count closely matching the effective spp of the reprojected frame.

In general, we see that the quality of the reprojected frame (middle column) matches the higher-spp comparison frame (right column) reasonably well also visually, although it typically exhibits some artifacts. The most prevalent artifacts are near the borders, which is due to the camera movement revealing new noisy data that have not yet been visible for long enough to have been smoothed out by the temporal accumulation. The uneven noise pattern after reprojection is also sometimes more noticeable (Fig. 7h) than the more evenly distributed noise in the purely path traced comparison (Fig. 7i), but overall we found the SSIM-based effective spp to describe the resulting visual quality more accurately than using RMSE, which tends to underestimate the effective spp.

4) NON-DIFFUSE MATERIALS

For the Sponza, Living Room and Classroom scenes, the effective spp of the reprojected glossy scene is, perhaps unexpectedly, much higher than in the diffuse case. In terms of visual quality, the fireflies are distributed more sporadically in the reprojected frames than in the higher-spp comparison frames, and the bilinear blur is visible in the reprojected frames to some extent as expected; otherwise, their quality is comparable. However, the scenes are overall much darker compared to the diffuse scenes, consequently smoothing out a lot of the texture and other fine details. This likely explains the significantly better results in terms of effective spp.

In the Classroom (Mirror) scene, the overall visual quality does not suffer due to the presence of the mirror. However, the reprojection (Fig. 7n) causes the objects in the mirror to be shifted by several pixels to the right compared to the purely path traced comparison (Fig. 7o); in terms of the effective spp, the quality drops from 60 spp to 57 spp. This happens because



(a) 1 spp (b) 1 spp reprojected (effective 20.03 spp) (c) 20 spp comparison

(d) 1 spp (e) 1 spp reprojected (effective 31.86 spp) (f) 32 spp comparison

(g) 2 spp (h) 2 spp reprojected (effective 44.43 spp) (i) 44 spp comparison

(j) 2 spp (k) 2 spp reprojected (effective 70.56 spp) (l) 71 spp comparison

(m) 4 spp (n) 4 spp reprojected (effective 57.04 spp) (o) 57 spp comparison

FIGURE 7. Frame comparison for the results in Section IV-B. Left column: left-eye frame with purely path traced *n* spp. Middle column: same left-eye frame obtained by spatiotemporally reprojecting the corresponding *n* spp right-eye frame onto an empty left eye and path tracing only the missing samples. Right column: purely path traced comparison frame with the spp count closely matching the effective spp of the reprojected frame.

for simplicity, our reprojection algorithm does not explicitly consider non-diffuse materials; reprojected samples are deemed valid or invalid based only on the differences in depth, world position, and shading normals. However, there are several sophisticated approaches for utilizing reprojection for reflections and even refractions, e.g., [33]–[35].

(p) 4 spp (q) 4 spp reprojected (effective 72.50 spp) (r) 72 spp comparison

(s) 8 spp (t) 8 spp reprojected (effective 101.12 spp) (u) 101 spp comparison

FIGURE 7. **(Continued:) Frame comparison for the results in Section IV-B. Left column: left-eye frame with purely path traced *n* spp. Middle column: same left-eye frame obtained by spatiotemporally reprojecting the corresponding *n* spp right-eye frame onto an empty left eye and path tracing only the missing samples. Right column: purely path traced comparison frame with the spp count closely matching the effective spp of the reprojected frame.**

TABLE 5. **Effective spp after stereo reprojection only (with stereo blending), as measured by SSIM. The average spp for each scene is taken over 50 frames, with the respective standard deviation also shown.**

	1 spp		2 spp		4 spp		8 spp		16 spp		32 spp	
	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std
Sponza	**2.57**	0.03	**4.92**	0.05	**9.62**	0.14	**19.04**	0.37	**37.56**	0.90	**73.96**	2.15
Sponza (Glossy)	**2.61**	0.04	**5.03**	0.09	**9.91**	0.21	**19.84**	0.57	**39.57**	1.13	**78.16**	2.24
Sponza (Moving Light)	**2.57**	0.06	**4.95**	0.15	**9.73**	0.29	**19.25**	0.58	**38.10**	1.36	**75.09**	3.26
Living Room	**2.49**	0.02	**4.78**	0.02	**9.47**	0.04	**18.89**	0.08	**37.70**	0.13	**75.34**	0.32
Living Room (Glossy)	**2.62**	0.02	**5.07**	0.05	**10.08**	0.11	**19.88**	0.20	**39.17**	0.27	**77.42**	0.50
Living Room (Moving Light)	**2.47**	0.02	**4.79**	0.04	**9.51**	0.05	**18.94**	0.08	**37.80**	0.11	**75.43**	0.33
Classroom	**2.48**	0.01	**4.70**	0.02	**9.21**	0.07	**18.22**	0.10	**35.98**	0.27	**70.53**	0.62
Classroom (Glossy)	**2.51**	0.02	**4.81**	0.03	**9.45**	0.07	**18.68**	0.12	**36.89**	0.26	**72.41**	0.53
Classroom (Mirror)	**2.47**	0.01	**4.69**	0.02	**9.17**	0.05	**18.10**	0.09	**35.65**	0.21	**69.53**	0.50
San Miguel	**2.40**	0.02	**4.71**	0.04	**9.30**	0.08	**18.32**	0.14	**35.99**	0.29	**70.28**	0.64
Scene Average ± Std	**2.52** ± 0.07		**4.85** ± 0.14		**9.54** ± 0.29		**18.92** ± 0.62		**37.44** ± 1.33		**73.81** ± 3.02	

TABLE 6. **Effective spp after spatiotemporal reprojection (with stereo blending), as measured by SSIM. The average spp for each scene is taken over 50 frames, with the respective standard deviation also shown.**

	1 spp		2 spp		4 spp		8 spp		16 spp		32 spp	
	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std
Sponza	**30.00**	1.87	**50.79**	3.66	**83.52**	7.39	**130.05**	13.45	**189.04**	21.88	**253.50**	31.68
Sponza (Glossy)	**50.98**	3.98	**94.04**	7.70	**168.11**	13.78	**289.09**	26.00	**477.92**	45.05	**760.22**	72.86
Sponza (Moving Light)	**28.45**	1.40	**48.45**	3.38	**80.01**	7.99	**125.34**	16.59	**184.31**	31.31	**251.78**	53.72
Living Room	**36.07**	0.94	**69.45**	1.93	**133.48**	4.23	**252.86**	8.97	**466.70**	18.27	**832.76**	34.68
Living Room (Glossy)	**58.55**	1.23	**100.65**	1.81	**175.84**	3.55	**310.65**	6.48	**547.06**	10.49	**943.91**	20.94
Living Room (Moving Light)	**34.26**	1.06	**65.51**	1.45	**125.27**	2.38	**235.77**	4.25	**430.98**	8.60	**756.19**	17.66
Classroom	**29.10**	0.71	**51.18**	1.65	**86.88**	3.99	**138.82**	8.52	**203.94**	15.69	**271.44**	24.26
Classroom (Glossy)	**36.22**	1.51	**62.48**	2.82	**104.97**	5.15	**169.65**	8.53	**255.95**	13.64	**356.31**	20.34
Classroom (Mirror)	**28.50**	0.74	**49.69**	2.01	**83.30**	4.95	**130.92**	11.10	**188.86**	20.72	**247.11**	32.35
San Miguel	**36.36**	1.15	**56.80**	2.47	**90.19**	5.03	**142.82**	9.72	**220.64**	17.22	**322.70**	27.19
Scene Average ± Std	**36.85** ± 10.14		**64.90** ± 18.59		**113.16** ± 36.00		**192.60** ± 72.17		**316.54** ± 145.46		**499.59** ± 285.15	

C. REPROJECTION WITH STEREO BLENDING

In Section IV-B we used reprojection for synthesizing another spatial viewpoint with path traced data from one viewpoint. Here, we instead assume that both viewpoints have already been path traced with *n* spp, but we leverage their redundancy by spatially cross-reprojecting between each viewpoint and

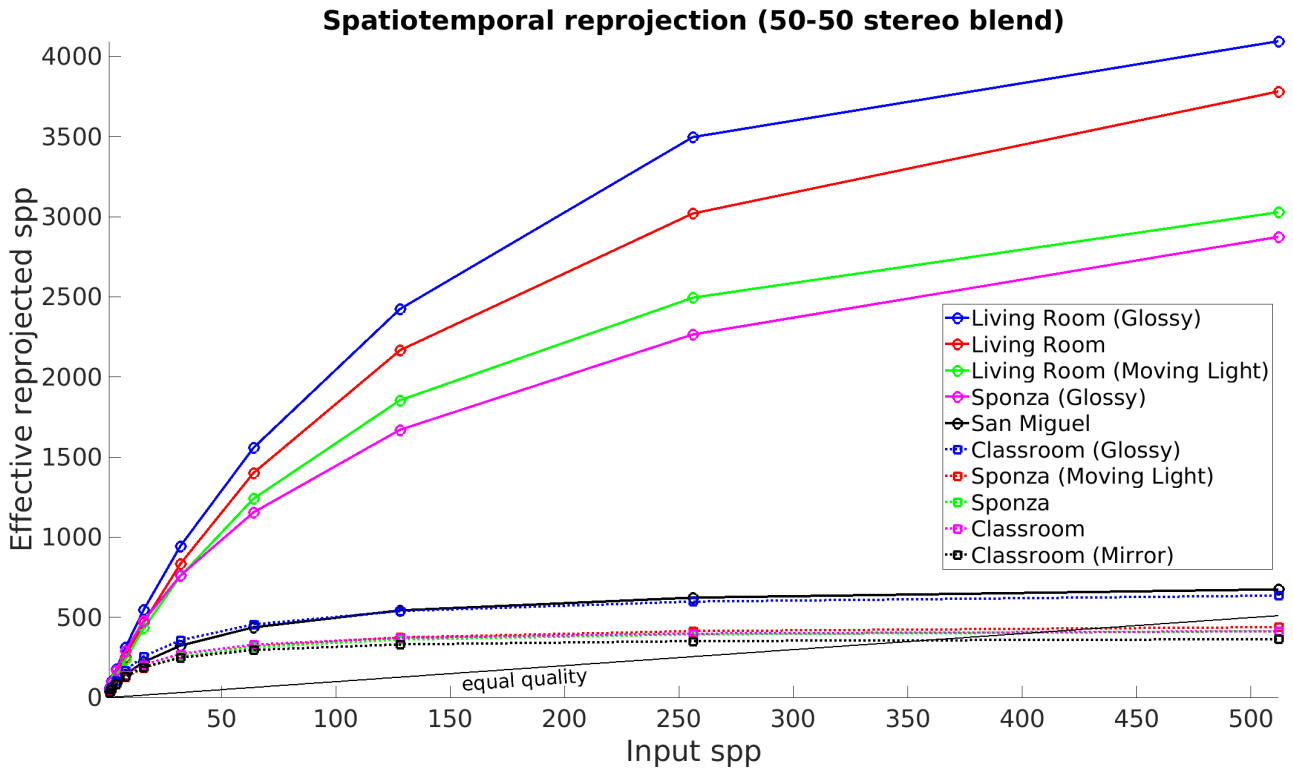


FIGURE 8. Average effective spp counts after spatiotemporal reprojection (with stereo blending) as a function of input spp. The results for 1–32 spp are also presented in Table 6.

then blending the reprojected pixels with the path traced pixels. As both viewpoints have an equal input quality of n spp, the blending is done with a 50-50 ratio.

The effective spp counts are evaluated as in Section IV-B, and the contributions are again separated into stereo reprojection, temporal reprojection, and the combined spatiotemporal reprojection. The temporal reprojection results are not affected by the stereo blending, hence they are equal to the results of Table 2; the results for stereo reprojection and spatiotemporal reprojection (for the left-eye target viewpoint) are presented in Table 5 and Table 6, respectively.

As we now reproject 1 spp data and also blend it with 1 spp data, the bilinear filtering improves the expected 2 spp quality after stereo reprojection to an effective 2.52 spp on average. Further, we get an average of 4.85 spp for 2 spp input, 9.54 spp for 4 spp input, 18.92 spp for 8 spp input, 37.44 spp for 16 spp input, and 73.81 spp for 32 spp input. As in the stereo reprojection done in Section IV-B, the results scale almost linearly over this range.

For the combined spatiotemporal reprojection with the 50-50 stereo blending, the average effective spp results are between 28.4–58.5 spp for 1 spp input, 48.4–100.7 spp for 2 spp input, 80.0–175.8 spp for 4 spp input, 125.3–310.6 spp for 8 spp input, 184.3–547.1 spp for 16 spp input, and 247.1–943.9 spp for 32 spp input. In other words, even the minimum improvement for 1 spp input is now 28-fold, as opposed to the 19-fold minimum improvement without stereo blending. On average, the effective spp is 47% higher than without stereo blending. This comes at the cost of path tracing n spp data for both viewpoints instead of doing it for only one of them. The scalability for higher-spp inputs is illustrated in Fig. 8; the behaviour is again similar to that observed in Fig. 5.

V. CONCLUSIONS

We presented a systematic evaluation of how the quality of noisy stereoscopic path traced data improves through spatiotemporal sample reprojection and temporal accumulation. We focused mainly on low-spp input data (1–32 spp), which is either currently possible to path trace in real time, or is expected to become feasible in the near future with more efficient hardware and software.

The results were presented in terms of the effective spp of the reprojected frames, demonstrating quality gains of an order of magnitude, while also saving 94–98% of the path tracing computations. We also showed that the quality improvement gained through stereo reprojection scales almost linearly on the 1–32 spp range, whereas for temporal reprojection the scaling is more sublinear. However, as the input spp count increases to the order of hundreds, the gains obtained through reprojection typically diminish quickly. Finally, we demonstrated how to achieve additional gains through stereo blending existing path traced data with

reprojected data, on average yielding a 47% higher effective spp than without blending.

Our evaluation focused on stereoscopic path traced data, but reprojection is also applicable in many other scenarios. For instance, various multi-view scenarios contain even more redundancy between the viewpoints, thus presenting an interesting context within which to further explore reprojection and its effectiveness. On the other hand, in fast-paced VR/AR applications, or in gaming in general, the movement within the scene can be highly unpredictable. Such applications call for robust reprojection methods that are able to quickly react and adapt to the rapidly changing environments; systematically evaluating the visual quality can also serve as an important tool in the development of such methods.

Finally, as the practically feasible real-time input spp count increases, it is interesting to consider the combination of reprojection and denoising filters, and at which point reprojection alone produces such accurate results that the following denoising filter can be significantly simplified or even completely omitted. Overall, our results can be used to analyze the effect of sample reprojection on image quality in denoising and anti-aliasing frameworks that take advantage of spatial or temporal coherence.

ACKNOWLEDGMENT

The authors would like to thank the creators of the original 3D models used in the experiments: F. Meinl for Crytek Sponza, Wig42 for Living Room, C. Seux for Classroom, and G. M. Leal Llaguno for San Miguel.

REFERENCES

[1] J. T. Kajiya, "The rendering equation," *ACM SIGGRAPH Comput. Graph.*, vol. 20, no. 4, pp. 143–150, Aug. 1986.

[2] S. Bako, T. Vogels, B. Mcwilliams, M. Meyer, J. Novák, A. Harvill, P. Sen, T. Derose, and F. Rousselle, "Kernel-predicting convolutional networks for denoising Monte Carlo renderings," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–14, Jul. 2017.

[3] M. Mäkitalo, P. Kivi, M. Koskela, and P. Jääskeläinen, "Reducing computational complexity of real-time stereoscopic ray tracing with spatiotemporal sample reprojection," in *Proc. 14th Int. Joint Conf. Comput. Vis., Imag. Comput. Graph. Theory Appl.*, vol. 1, 2019, pp. 367–374.

[4] I. E. Sutherland, R. F. Sproull, and R. A. Schumacker, "A characterization of ten hidden-surface algorithms," *ACM Comput. Surv.*, vol. 6, no. 1, pp. 1–55, Mar. 1974.

[5] T. Whitted, "An improved illumination model for shaded display," *ACM SIGGRAPH Comput. Graph.*, vol. 13, no. 2, p. 14, Aug. 1979.

[6] R. L. Cook, T. Porter, and L. Carpenter, "Distributed ray tracing," *ACM SIGGRAPH Comput. Graph.*, vol. 18, no. 3, pp. 137–145, 1984.

[7] S. Badt, "Two algorithms for taking advantage of temporal coherence in ray tracing," *Vis. Comput.*, vol. 4, no. 3, pp. 123–132, May 1988.

[8] S. J. Adelson and L. F. Hodges, "Generating exact ray-traced animation frames by reprojection," *IEEE Comput. Graph. Appl.*, vol. 15, no. 3, pp. 43–52, May 1995.

[9] B. Walter, G. Drettakis, and S. Parker, "Interactive rendering using the render cache," in *Rendering Techniques'99*, vol. 10, D. Lischinski and G. Larson, Eds. Granada, Spain: Springer-Verlag/Wien, Jun. 1999, pp. 235–246. [Online]. Available: https://hal.inria.fr/inria-00510073

[10] I. Wald and P. Slusallek, "State of the art in interactive ray tracing," in *Proc. State Art Rep., EUROGRAPHICS*, vol. 2001, 2001, pp. 21–42.

[11] D. Nehab, P. V. Sander, J. Lawrence, N. Tatarchuk, and J. R. Isidoro, "Accelerating real-time shading with reverse reprojection caching," *Graph. Hardw.*, vol. 41, pp. 61–62, Aug. 2007.

[12] D. Scherzer, S. Jeschke, and M. Wimmer, "Pixel-correct shadow maps with temporal reprojection and shadow test confidence," in *Proc. 18th Eurographics Conf. Rendering Techn.*, 2007, pp. 45–50.

[13] M. Koskela, K. Immonen, M. Mäkitalo, A. Foi, T. Viitanen, P. Jääskeläinen, H. Kultala, and J. Takala, "Blockwise multi-order feature regression for real-time path-tracing reconstruction," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1–14, Nov. 2019.

[14] C. Schied, C. Peters, and C. Dachsbacher, "Gradient estimation for real-time adaptive temporal filtering," *Proc. ACM Comput. Graph. Interact. Techn.*, vol. 1, no. 2, pp. 1–16, 2018.

[15] C. Schied, A. Kaplanyan, C. Wyman, A. Patney, C. R. A. Chaitanya, J. Burgess, S. Liu, C. Dachsbacher, A. Lefohn, and M. Salvi, "Spatiotemporal variance-guided filtering: Real-time reconstruction for path-traced global illumination," in *Proc. High Perform. Graph.*, Jul. 2017, p. 2.

[16] M. Mara, M. McGuire, B. Bitterli, and W. Jarosz, "An efficient denoising algorithm for global illumination," in *Proc. High Perform. Graph.*, Jul. 2017, p. 78.

[17] B. Karis, "High-quality temporal supersampling," *Adv. Real-Time Rendering Games, SIGGRAPH Courses*, vol. 1, no. 10, p. 1145, 2014.

[18] L. Yang, S. Liu, and M. Salvi, "A survey of temporal antialiasing techniques," *STAR*, vol. 39, no. 2, pp. 607–621, 2020.

[19] A. D. Corso, M. Salvi, C. Kolb, J. R. Frisvad, A. Lefohn, and D. Luebke, "Interactive stable ray tracing," in *Proc. High Perform. Graph.*, Jul. 2017, pp. 1–10.

[20] D. Scherzer, L. Yang, O. Mattausch, D. Nehab, P. V. Sander, M. Wimmer, and E. Eisemann, "Temporal coherence methods in real-time rendering," *Comput. Graph. Forum*, vol. 31, no. 8, pp. 2378–2408, 2012.

[21] M. Zwicker, W. Jarosz, J. Lehtinen, B. Moon, R. Ramamoorthi, F. Rousselle, P. Sen, C. Soler, and S.-E. Yoon, "Recent advances in adaptive sampling and reconstruction for Monte Carlo rendering," *Comput. Graph. Forum*, vol. 34, no. 2, pp. 667–681, 2015.

[22] A. J. Woods, T. Docherty, and R. Koch, "Image distortions in stereoscopic video systems," in *Proc. SPIE*, vol. 1915, pp. 36–49, Sep. 1993.

[23] R. Allison, "Analysis of the influence of vertical disparities arising in toed-in stereoscopic cameras," *J. Imag. Sci. Technol.*, vol. 51, pp. 317–327, Jul./Aug. 2007.

[24] S. J. Adelson and L. F. Hodges, "Stereoscopic ray-tracing," *Vis. Comput.*, vol. 10, no. 3, pp. 127–144, Mar. 1993.

[25] J. D. Ezell and L. F. Hodges, "Some preliminary results on using spatial locality to speed up ray tracing of stereoscopic images," *Proc. SPIE*, vol. 1256, pp. 298–307, Sep. 1990.

[26] H. H. S. Ip, K. C. K. Law, and G. K. P. Fung, "Epipolar plane space subdivision method in stereoscopic ray tracing," *Vis. Comput.*, vol. 13, no. 6, pp. 247–264, Aug. 1997.

[27] G. Rosado, "Motion blur as a post-processing effect," in *GPU Gems 3*. Reading, MA, USA: Addison-Wesley, 2007, ch. 27.

[28] M. Andersson, B. Johnsson, J. Munkberg, P. Clarberg, J. Hasselgren, and T. Akenine-Möller, "Efficient multi-view ray tracing using edge detection and shader reuse," *Vis. Comput.*, vol. 27, nos. 6–8, p. 665, 2011.

[29] B. Fraboni, J.-C. Iehl, V. Nivoliers, and G. Bouchard, "Adaptive multi-view path tracing," in *Proc. 30th Eurographics Symp. Rendering*, 2019, pp. 1–10.

[30] S. Bhonde and M. Shanmugam. (2018). *Turing Multi-View Rendering in VRWorks*. [Online]. Available: https://devblogs.nvidia.com/turing-multi-view-rendering-vrworks

[31] M. Koskela, A. Lotvonen, M. Mäkitalo, P. Kivi, T. Viitanen, and P. Jääskeläinen, "Foveated real-time path tracing in visual-polar space," in *Proc. 30th Eurographics Symp. Rendering*, 2019, pp. 39–50.

[32] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[33] G. Lochmann, B. Reinert, T. Ritschel, S. Müller, and H.-P. Seidel, "Real-time reflective and refractive novel-view synthesis," in *Proc. VMV*, 2014, pp. 9–16.

[34] H. Zimmer, F. Rousselle, W. Jakob, O. Wang, D. Adler, W. Jarosz, O. Sorkine-Hornung, and A. Sorkine-Hornung, "Path-space motion estimation and decomposition for robust animation filtering," *Comput. Graph. Forum*, vol. 34, no. 4, pp. 131–142, 2015, pp. 131–142.

[35] N. Xie, L. Wang, and P. Dutré, "Reflection reprojection using temporal coherence," *Vis. Comput.*, vol. 34, no. 4, pp. 517–529, Apr. 2018.

MARKKU J. MÄKITALO received the Ph.D. degree in signal processing from the Tampere University of Technology (TUT), in 2013. He is currently a Postdoctoral Researcher with Tampere University. His research interests include photorealistic real-time multi-view rendering, low-latency video compression, and immersive rendering technologies, such as light field rendering.

PETRUS E. J. KIVI received the master's degree in mathematics from Tampere University, in 2020. He is currently a Project Researcher with Tampere University. His research interests include stereoscopic rendering and point cloud rendering.

PEKKA O. JÄÄSKELÄINEN is an Assistant Professor. He has worked on heterogeneous platform customization and programming since the early 2000s. He has published over 90 academic articles in journals and conferences, and is an active contributor to various heterogeneous parallel platform related open source projects. In addition to his ongoing research on methods and tools to reduce the engineering effort involved in design and programming of heterogeneous platforms, he is interested in the next generation programmable graphics architectures for photorealistic real-time rendering in the context of small form factor VR/AR products of the future.